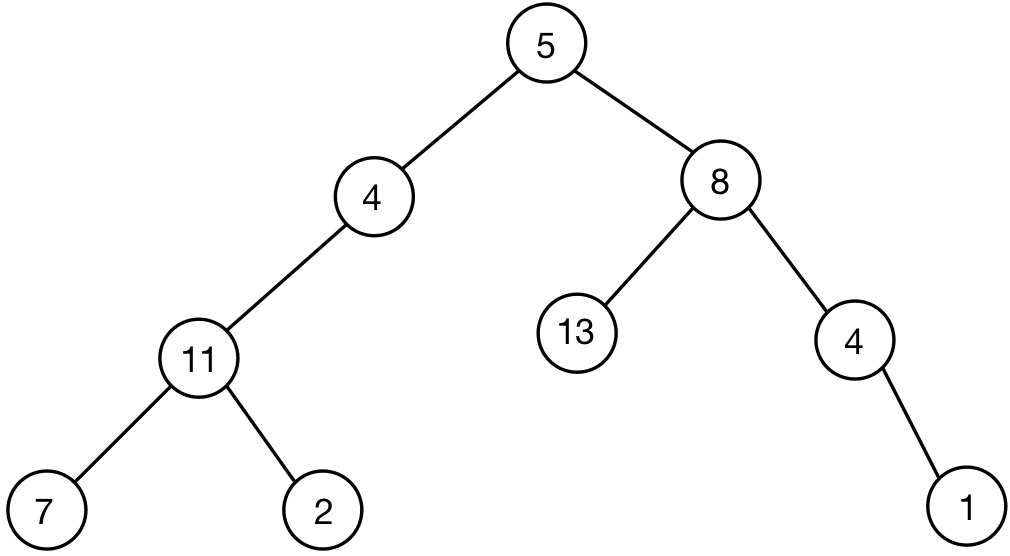
Path Sum 2 (LeetCode 113)

*Description:*

Given a binary tree and a sum, find all root - to - leaf paths where each path’s sum equals the given sum.

*For example:*

Given the below binary tree and sum = 22:



*Code:*

*/\*\**

*\* Definition for a binary tree node.*

*\* struct TreeNode {*

*\*     int val;*

*\*     TreeNode \*left;*

*\*     TreeNode \*right;*

*\*     TreeNode(int x) : val(x), left(NULL), right(NULL) {}*

*\* };*

*\*/*

*class Solution {*

*public:*

*vector<vector<int>> pathSum(TreeNode\* root,*

*int sum) {*

*vector<vector<int>> result;*

*vector<int> vec;*

*intermediatePathSum(root, sum, result, vec);*

*return result;*

*}*

*/\**

*\*\* Step 1:*

*\*\* Push the current node into the vec as long as it is not the leaf node,*

*\*\* if the current node has no left/right child node, and left\_sum equals to 0,*

*\*\* then this routine would satisfy the condition of path sum.*

*\*\* Under all other conditions, just skip and return to the upper level.*

*\*\* Step 2:*

*\*\* Recursively run the process on the left node.*

*\*\* Step 3:*

*\*\* Recursively run the process on the right node.*

*\*\* Step 4:*

*\*\* Pop out the current node out of the vec and finish the current phase.*

*\*/*

*void intermediatePathSum(TreeNode\* root,*

*int sum,*

*vector<vector<int>> &result,*

*vector<int> &vec) {*

*if (root == NULL) {*

*return;*

*}*

*vec.push\_back(root->val);*

*int sum\_left = sum - root->val;*

*if (root->left == NULL*

*&& root->right == NULL*

*&& sum\_left == 0) {*

*result.push\_back(vec);*

*}*

*intermediatePathSum(root->left, sum\_left, result, vec);*

*intermediatePathSum(root->right, sum\_left, result, vec);*

*vec.pop\_back();*

*}*

*};*